DØ PPDG Year 3 plans

Table of Contents

DØ Team

1 Introduction

The DØ-PPDG project is the core of the D0Grid (now SAMGrid). It's continuing mission to provide truly distributed (in the global sense) computing for the Run II experiments. We leverage SAM – a mature, advanced data management system, and add services for job and information management, while collaborating with computer scientists, most notably, University of Wisconsin. Our principal deliverable in SAMGrid is secure, reliable execution of structured, globally distributed jobs, with sufficient provision of monitoring, both at run time and historically.

During the first two years of our participation in the collaboration, we have delivered the first version of JIM (the Job and Information Management of SAMGrid), where we integrated the power of Condor-G for Grid job brokering with that of SAM for global data movement. Our global scheduler dispatches jobs based on the amount of data cached at participating sites. In addition, web-based monitoring of the whole system and individual jobs was provided. Presently, JIM V1 is in the process of deployment. [detailed status available elsewhere]

2 PPDG Year 3 Plans

During the third year of the collaboration, we plan to stabilize JIM v1, expose it to high operational load. We will define and deliver Version 2 of JIM, as well as prototype an operational support model for the (remote) SAMGrid installations.

Planned general deliverables for SAM/GRID

- Devise a viable model for job input and output transfers of executables, associated flat files other than collider data (e.g., control files and calibrations), log and error outputs from the consumer, other consumer outputs which do not fit the model of data tiers
- Support Monte Carlo job distribution via the brokering service at a level of sophistication which replaces the current manual job distribution (i.e., uses the capacities of the production centers and implements the job priorities as input to the Monte Carlo request system
- Supply a monitoring and information service for servers and jobs, which will include better mechanisms for filtering and archiving than the current flat file logs, and allow retrieval of basic performance metrics
- Supply functional 'virtual organization' management system, either as further development of our current tools, or adoption of new technologies as they are available.
- Support reconstruction/analysis job distribution at the level of current functionality (manual choice of site), but making use of the job input/output model above
- Begin to understand brokering issues for reconstruction and analysis jobs.

The following developments will have to take place in JIM:

- Resource description for MC environments has to be frozen and published, using the JIM-adopted Condor ClassAd framework. Presently, a lot of ideas have been circulated and a prototypical implementation is available. The same applies to reconstruction.
- The DMZ between the Grid and the Fabric has to be clearly defined. One of the novelties of JIM's design is the use of the XML database on the border between the Grid and the Fabric. We will need to expose this topic to a broader Grid community so as to stimulate discussions and possibly standardize the interface between the Fabric and the Grid.
- The XML-based logger has to be fully developed. Unlike the Logging and Book-keeping service from EDG WP1, our logging service is an important concept which will underlie historical data mining for both data transfers and jobs, and thus provide accounting at various levels.
- Evaluation of Web services must also be complete by the end of year 3. While any particular grid-like system like SAM can do without web services, it is becoming increasingly clear that it is impossible to combine diversity in grid solutions with grid interoperability without a common language to describe

services, and WSDL is the *de facto* such leader. Thus, unless the Grid community in general and Run II experiments in particular want to entrench into middleware consisting of proprietary (GTK 2) implementations of obscure, often criticized protocols (GRAM, GridFTP), we must arm ourselves with proper generalizations and describe our system in a language like WSDL.

The above work will keep the two PPDG FTE's (with helper students) well occupied. In addition, more work will be required to SAM. However, we envision more intense evolution of this data handling system towards the Grid. Thus, the evolution of SAM needs to be linked more tightly with PPDG activities and we will make according proposals for years 4 and 5 below.

2.1 Action Items from '03 Questionnaires

We continue to work closely with the Condor team to develop and test the three-tier condor architecture, and test the matchmaking features provided for JIM by the Condor team. We ask that these efforts continue to be high priority for the Condor team. An installation of a JIM Gateway at UW Madison is in progress and this will facilitate continued cooperation between the JIM and Condor teams, and further understanding, and testing of the relevant software.

2.2 The JIM Project JIM

We are now in the process of deploying the first production version of the JIM product for DØ and CDF. Success of the software is measured by first providing the experiments with the functionality already existing within each experiment to accomplish their computing tasks. The next step is enabling the automated job management features which provide transparent access to new resources, and reduce the manpower requirements needed to manage these activities.

The following list of deliverables are items the team, DØ and CDF, feel are needed to make the system fully functional for the Run II experiments. The JIM system is planned to function in three main modes of operation and job manager interfaces are being provided for 1) Monte Carlo production, 2) production processing like reconstruction, and 3) user analysis. Job scheduling functionality will be phased in over the coming months as follows: Phase 1) Provide existing functionality with manual decisions for execution site selection, Phase 2) Enable true brokering of jobs using execution site resource advertisement and the condor match making to rank jobs, and Phase 3) Manage structured jobs which are split intelligently and the components managed in parallel.

2.2.2 Deliverables

We rank the following list of deliverables into three categories, the first are high priority which we plan to accomplish in the next year, second are lower priority tasks, and the third set are important tasks but for which we rely on the other teams for completion. Estimates of D0-PPDG team FTE-Months required to accomplish each task are given in [].

2.2.2.1 Highest priority Deliverables [Est. D0-PPDG team FTE-Months: 30]

1. Deployment of JIM for DØ Sites (execution, submission, monitoring) **[6 FTE Months]**
    a. Initial DØ sites: GridKa, Wuppertal, Lancaster, IC,U. Michigan, and UW Madison.
    b. Extended DØ site list: Lyon, UTA, NIKHEF, RAL, Manchester, and Prague.
    c. Extended DØ site list for submission (potentially any D0 site)
2. Deployment of JIM for CDF Sites (execution, submission, monitoring) **[4 FTE Months]**
    a. Initial installation: Mini-CAF, and Scotgrid (Glasgow + Edinbourg).
    b. Second round: sites may include GridKa,, Oxford, Trieste, Toronto, Kyungpook S. Korea, TexasTech, and UCSD.
3. Support for JIM for both CDF and DØ. **[4 FTE Months]**
4. Resolve issues with input and output sandbox . **[2 FTE Months]**

5. Improve installation framework; Reduce installation time by clarifying the configuration process (interaction with the installer) and developing more intelligent guesses for the defaults. Packaging and code distribution with ups/upd needs to be re-evaluated. Improve upgrade procedure. **[2 FTE Months]**
6. Additional troubleshooting tools + test of "sanity" procedure (e.g. after installation). **[2 FTE Months]**
7. Hardening of the system will come with testing and production experience. Extensive load testing with a test harness application will be done and Upgrading to new/more robust versions of globus and condor should help. **[2 FTE Months]**
8. Further deliverables from collaboration with Condor team including continued improvement of the 3-tier architecture, improved authentication to provide true 3-tier functionality; retrieval of output from execution site; and management ownership of job. Additional debugging as problems are observed. **[2 FTE Months]**
9. Logging service is extremely important and could be ready in 6 months if we start now. The plan is to leverage existing systems such as netlogger if possible. We would like an XML based system with support for both reliable and unreliable message delivery. It needs to have a hierarchical model to support required scalability. **[6 FTE Months]**

2.2.2.2 Lower priority deliverables [Est. D0-PPDG team FTE-Months: 22]

1. Configuration management improvements although the XML framework that we now have is probably adequate and enables advertisement of the configuration through a web interface. **[4 FTE Months]**
2. Monitoring service; the current MDS implementation is probably not adequate and we will explore alternatives in the areas of web services. Also, the monitoring requires proper interfaces to the fabric. We will look at existing technologies where possible, such as Condor Hawkeye. **[4 FTE Months]**
3. JDL: Job definition language for structured jobs, defining the issues of what is a job, how is it parsed, and how is it related to the condor. **[6 FTE Months]**
4. Evaluation of Web services; as condor and Globus transition to move web services we will be forced to use these technologies and develop web services for SAMGrid. **[4 FTE Months]**
5. Gathering statistics on relevant metrics to understand what parameters are minimized/maximized using the (SAM-)Grid paradigm to computation. **[4 FTE Months]**

2.2.2.3 Deliverables from other teams on which we rely [Est. D0-PPDG team FTE-Months: unknown]

1. Workflow management needs a job manager interface for runjob. This work is done in cooperation with CMS, FNAL-CD, UK players in runjob (a.k.a. Shahkar project). We would also like to investigate redundancy/fail-over models. **[2-4 FTE Months]**
2. Follow evolution of SAM such as SRM used in general cache and storage strategy for SAM. **[unknown]**
3. We would like to understand the solution for VO management in the next 6 months. This will be done in conjunction with one or more of following efforts: The existing JIM VOMS (written by Gabriele Garzoglio), the EDG VOMS project, or FNAL/USCMS/SDSS VOX project. Possible implementation. **[2-4 FTE Months]**
4. Return output file from jobs using a transient sam storage location being discussed in the core SAM team. **[unknown]**
5. Distributed Replica Catalogues possibly using a "standard" Replica Location Service (RLS if available. **[unknown]**
6. We will continue work on common areas of the Glue Schema and implement where relevant. **[unknown]**

2.2.3 Milestones

7/1/2003: 1) Finish installation of JIM at initial sites; 2) Begin load testing; 3) Exercise analysis; 4) Resolve issues with user input and output sandbox management; 5) Evaluate VO management options; 6) begin building additional load testing procedures.

10/1/2003: 1) Jim V1.1 release; 2) Installation framework ready; 3) Use of VDT for release; 4) Fully functional configuration management; 5) Improved monitoring; 6) Full three-tier condor implementation; 7)stdout and stderr returned through condor spool; 8) Second round of installs; 9) Initial logging service features; 10)Establish an operational support model; 11) Integration of VO management tools.

1/1/2004: 1) JIM V1.2 release; 2) Full logging service features ready; 3) Initial management of structured jobs; 4) Additional scheduling functionality; 5) Augmented resource advertisement; 6) Begin further evaluation of Web Services; 7) integration w/ SAM transient file transfer features if ready 8) Round 3 deployment, 9) Updated support features; 10) improved reliability.

4/1/2004: 1) JIM V1.3 release; 2) Move toward Globus and Condor Web Services implementations (pending their status); 3) Evaluating VO options (pending availability); 4) Full accounting statistics available; 5) Improved support and reliability; 6) Contingency for previous missed milestones.

7/1/2004: 1) JIM V2; 2) Begin exploring distributed replica catalog solutions; 3) More work on web services.

2.2.4 Participants

Igor Terekhov - Technical Team Lead

Lee Lueking – D0 Team Lead

Gabriele Garzoglio

Andrew Baranovski

Vijay Murthi – CS Masters student through contract with U. Texas Arlington

Parag Mhashilkar – CS Masters student through contract with U. Texas Arlington

2.2.5 Dependencies and Critical Partnerships

The JIM project has close dependencies on the Condor team especially through the Condor matchmaking, condor three-tier architecture and , VDT. We rely on the use of the Globus tool kit. We depend on the UK GridPP effort especially through RunJob and other contributions they make to the JIM deployment and testing. We have tight integration with the Fermilab CD Core SAM project and Run II experiments.

2.2.6 Issues and Concerns

The JIM team is faced with many large and challenging tasks in several areas. There are many dependencies on software out of our immediate control and coupled to other teams. Support for CDF is a concern and costs significant time and effort, although we like seeing the common solutions for the Run II experiments that are emerging

3 PPDG Year 4 and 5

Generally, future work will proceed along two avenues: major feature enrichments (robustness, ease of use, better system control) as well as interoperability with other grids, if any.

As far as JIM is concerned, we propose:

- Manage *structured* jobs at the global level, including decomposition and recombination of job's fragments according to the job details known to the scheduler. Presently, this cumbersome job is done manually by e.g. MC request executor. Ultimately, such decomposition should take place dynamically, i.e. as more resources become available. Some efforts in this area are undertaken at the local cluster level, moving it to the grid level will present new challenges.
- Tolerance to entire grid cluster/site failure, with re-routing of the corresponding job fragments to new resources.
- The above two may collectively be called workflow management, but one should not confuse intra-cluster management with that at the grid level.

For inter-operability, we intend to pursue studies of the Glue schema and Job description languages. Both need to be looked at from a broader prospective that defining e.g. standard UML diagram or a standard language. We are more inclined to converging on a framework that allows to process data representing these entities.

As far as SAM per se is concerned, in addition to the previously mentioned enhancements, we will need the following:

Entanglement of all meta-data into a single Oracle database has to change. Some data items are critical for the entire collaboration and must be conceptually centralized. Other items are of highly transient nature and hinder deployment of large distributed caches at distributed sites. We will likely need splitting of the schema, decentralization of the catalog, moving some of the bookkeeping services into the logger (local and/or central), as well as replacement of select services (such as replica catalogue) with externally developed, more standard ones.

It is very likely that a single person will be continuously required for the above evolution of SAM towards JIM, in order for the overall success of D0-PPDG.

4 Background

4.1 Summary Table from PPDG proposal

This is the summary table from the PPDG proposal. Pluses ("+") have been placed in the Year 3 column to indicate D0 PPDG activities. The full proposal is available at http://lbnl2.ppdg.net/docs/scidac01_ppdg_public.doc

Project Activity	Experiments	Yr1	Yr2	Yr3
CS-1 Job Description Language – definition of job processing requirements and policies, file placement & replication in distributed system.				
P1-1 Job Description Formal Language	D0, CMS	X		+
P1-2 Deployment of Job and Production Computing Control	CMS	X		+
P1-3 Deployment of Job and Production Computing Control	ATLAS, BaBar, STAR		X	
P1-4 Extensions to support object collections, event level access etc.	All			X
CS-2 Job Scheduling and Management - job processing, data placement, resources discover and optimization over the Grid				
P2-1 Pre-production work on distributed job management and job placement optimization techniques	BaBar, CMS, D0	X		+
P2-2 Remote job submission and management of production computing activities	ATLAS, CMS, STAR, JLab		X	+
P2-3 Production tests of network resource discovery and scheduling	BaBar		X	
P2-4 Distributed data management and enhanced resource discovery and optimization	ATLAS, BaBar			X
P2-5 Support for object collections and event level data access. Enhanced data re-clustering and re-streaming services	CMS, D0			X

CS-3 Monitoring and Status Reporting				
P3-1 Monitoring and status reporting for initial production deployment	ATLAS	X		
P3-2 Monitoring and status reporting – including resource availability, quotas, priorities, cost estimation etc	CMS, D0, JLab	X	X	+
P3-3 Fully integrated monitoring and availability of information to job control and management.	All		X	X
CS-4 Storage resource management				
P4-1 HRM extensions and integration for local storage system.	ATLAS, JLab, STAR	X		
P4-2 HRM integration with HPSS, Enstore, Castor using GDMP	CMS	X		
P4-2 Storage resource discovery and scheduling	BaBar, CMS		X	
P4-3 Enhanced resource discovery and scheduling	All			X
CS-5 Reliable replica management services				
P5-1 Deploy Globus Replica Catalog services in production	BaBar, JLab	X		
P5-2 Distributed file and replica catalogs between a few sites	ATLAS, CMS, STAR, JLab	X		
P5-3 Enhanced replication services including cache management	ATLAS, CMS		X	
CS-6 File transfer services				
P6-1 Reliable file transfer	ATLAS , BaBar, CMS, STAR, JLab	X		
P6-2 Enhanced data transfer and replication services	ATLAS, BaBar, CMS, STAR, JLab		X	
CS-7 Collect and document current experiment practices and potential generalizations	All	X	X	X